# **Project Documentation: Digital Grievance Redressal System**

## **Overview**

The Digital Grievance Redressal System is a web-based application aimed at simplifying the process for citizens to submit and track complaints to local government bodies. The system also enables administrators to manage grievances efficiently, ensuring transparency and accountability.

## **Project Structure**

### **Frontend Structure**

1. **Login/Registration Page**
   1. Citizen and Admin login.
   2. Registration form for citizens.
2. **User Dashboard**
   1. Grievance submission form.
   2. List of submitted grievances with their statuses.
   3. Grievance tracking functionality.
3. **Admin Panel**
   1. View all submitted grievances.
   2. Categorize grievances by type and urgency.
   3. Update grievance status.
   4. Analytics dashboard.

### **Backend Structure**

1. **APIs for User Authentication**
   1. User registration, login, and token-based authentication.
2. **APIs for Grievance Management**
   1. Grievance submission.
   2. Status update.
   3. Fetch grievance details for users and admins.
3. **Notification Service**
   1. Email/SMS notifications for grievance submission, status updates, and resolution.

### **Database Structure**

1. **Users Table**
   1. user\_id (Primary Key)
   2. name
   3. email
   4. password
   5. role (Citizen/Admin)
2. **Grievances Table**
   1. grievance\_id (Primary Key)
   2. user\_id (Foreign Key)
   3. category
   4. description
   5. status (Pending/In Progress/Resolved)
   6. created\_at
   7. updated\_at
3. **Notifications Table**
   1. notification\_id (Primary Key)
   2. user\_id (Foreign Key)
   3. message
   4. sent\_at

## **Features Documentation**

### **Citizen Features**

1. **User Registration and Login**
   1. Citizens can register with details like name, email, and password.
   2. Login functionality with password hashing and validation.
2. **Grievance Submission**
   1. Submit grievances with the following fields:
      1. Category (e.g., Sanitation, Road Maintenance, Public Safety).
      2. Description of the issue.
      3. Option to upload images/documents.
3. **Grievance Tracking**
   1. View all submitted grievances in a list format.
   2. Check the status of each grievance (Pending, In Progress, Resolved).
4. **Notifications**
   1. Receive email/SMS alerts for grievance submission, status changes, and resolution.

### **Admin Features**

1. **Admin Login**
   1. Secure admin login with role-based access control.
2. **View and Manage Grievances**
   1. View all grievances submitted by citizens.
   2. Filter grievances by category, status, or submission date.
   3. Assign grievances to specific departments (optional feature).
3. **Update Grievance Status**
   1. Change the status of grievances to Pending, In Progress, or Resolved.
4. **Analytics Dashboard**
   1. Display total grievances submitted, resolved, and pending.
   2. Visual representation using charts (e.g., pie charts, bar graphs).
5. **Notifications**
   1. Send status updates to citizens via email/SMS.

## **Technical Details**

### **Frontend**

* Framework: React.js or Vue.js (optional: plain HTML/CSS/JavaScript).
* CSS Framework: Tailwind CSS/Bootstrap for styling.

### **Backend**

* Framework: Flask/Django (Python) or Express.js (Node.js).
* Authentication: JWT-based token system.
* Email/SMS API Integration: SendGrid (email) or Twilio (SMS).

### **Database**

* Relational Database: MySQL or PostgreSQL.
* ORM: SQLAlchemy (Python) or Sequelize (Node.js).

### **Deployment**

* Hosting: AWS, Heroku, or DigitalOcean.
* Version Control: GitHub/GitLab.

## **Development Workflow**

1. **Frontend Development**
   1. Build static pages for login, grievance submission, and dashboard.
   2. Add dynamic functionality using API integration.
2. **Backend Development**
   1. Set up database and build RESTful APIs.
   2. Implement authentication and role-based access control.
3. **Testing**
   1. Unit testing for APIs and frontend components.
   2. End-to-end testing to ensure smooth user experience.
4. **Deployment**
   1. Deploy the app and database on a cloud platform.
   2. Set up domain and SSL for secure access.

## **Future Enhancements**

1. **Multi-Language Support:** Add support for Nepali and other regional languages.
2. **Mobile App Integration:** Create a mobile app for easier access.
3. **AI Chatbot:** Use AI to assist users in submitting grievances or tracking issues.
4. **Department-Specific Dashboard:** Enable different departments to manage their own grievances.
5. **Anonymous Grievances:** Allow citizens to submit grievances anonymously if desired.

This documentation should provide you with a clear roadmap to build the project. Let me know if you need help with any specific section!